

09/879,838

**REMARKS**

The present Response is submitted in reply to the Official Action mailed July 6, 2004, and the Applicant respectfully requests entry of the amendments and arguments presented herein before reconsideration of this application.

Claims 1 through 14 are presently pending in the Application and the Examiner has rejected claims 1 through 14 under 35 U.S.C. 112 and under 35 U.S.C. 103 over prior art cited by the Examiner. The Examiner has also objected to the Specification for use of the trademark "JAVA" in a manner that does not fully represent the proprietary nature of the term as a trademark.

First considering the use of the trademark "JAVA", the Applicant has amended the Specification at page 5, line 21 through page 6, line 2, to fully acknowledge the status of the term JAVA™ as a trademark of SUN Microsystems Inc., and to indicate that the trademark JAVA™ would thereafter be referred to in the Specification by the term "Java". This approach to indicating the trademark nature of JAVA™ was adopted as compliant with previously accepted practice in patent applications and patents and because the term "Java" occurs multiple times upon virtually every page of the Specification, thereby providing appropriate notice of the status of JAVA™ as a trademark while avoiding a complete rewriting of the Specification. The Applicant therefore respectfully requests that the Examiner reconsider and withdraw the objection to the Specification as regards the use therein of the trademark JAVA™.

Next considering the rejections of claims under 35 U.S.C. 112, the Examiner has rejected claims 1, 8-10 and 13 under 35 U.S.C. § 112, second paragraph, as being indefinite for the reasons noted in the official action. More specifically, the Examiner objects generally to the terms "one or more" and "at least one of" in claims 1, 8, 9, 10 and 13 as indefinite and to the term "instrumentation or modifying" in claims 8 and 10 as indefinite.

The Applicant respectfully disagrees with the Examiner's holding that the terms "one or more" and "at least one of", as used in claims 1, 8, 9, 10 and 13, render the claims indefinite. The terms "one or more" and "at least one of" are long and commonly accepted terms in claim drafting practice and it has long been held, both in the Patent Office and in court decisions, that the terms "one or more" and "at least one of" are well defined and well and clearly understood and that the correct use of these terms, as in claims 1, 8, 9, 10 and 13, do not render claims indefinite. The Applicant therefore respectfully traverses the Examiner's rejections of the claims under 35 U.S.C.

09/879,838

112 on the grounds that the terms "one or more" and "at least one" render the claims indefinite and reserves this issue for later discussion or appeal, if necessary.

In order to advance the prosecution of the present Application and claims, however, the Applicant submits the above claims amendments which amend the claims in a manner to avoid use of the terms "one or more" and "at least one". It is the position of the Applicant, however, that the offered amendments are directed solely at overcoming the raised indefiniteness rejection(s) and are not directed at distinguishing the present invention from the art of record in this case and do not alter, modify or limit the scope or meaning of the claims as originally written and intended.

It will also be noted that the Applicant has also made certain amendments to the claims to address certain grammatical informalities noticed in the claims during the above amendments. Again, these amendments are directed solely to addressing certain grammatical informalities and are not directed at distinguishing the present invention from the art of record in this case and do not alter, modify or limit the scope or meaning of the claims as originally written and intended.

The rejected claims are accordingly amended, by the above claim amendments, and the presently pending claims are now believed to particularly point out and distinctly claim the subject matter regarded as the invention, thereby overcoming all of the raised § 112, second paragraph, rejections.

Next, considering the rejections of the claims over the prior art cited by the Examiner, the Examiner has rejected claims 1 and 8-13 are rejected under 35 U.S.C. § 103(a) as being unpatentable over U.S. Patent No. 6,314,558 to Angel et al. for BYTE CODE INSTRUMENTATION, hereafter referred to as "Angel et al. '558", in view of U.S. Patent No. 6,026,237 to Berry et al. for SYSTEM AND METHOD FOR DYNAMIC MODIFICATION OF CLASS FILES, hereafter referred to as Berry et al. '237. The Examiner has also rejected claims 2-7 and 14 under 35 U.S.C. § 103(a) as being unpatentable over Angel et al. '558 in view of U.S. Patent No. 6,405,367 to Bryant et al. for APPARATUS AND METHOD FOR INCREASING THE PERFORMANCE OF JAVA PROGRAMS RUNNING ON A SERVER, hereafter referred to as "Bryant et al. '367". The Applicant acknowledges and respectfully traverses all of the raised obviousness rejections in view of the following remarks.

First, considering the present invention as recited in independent claims 1 and 13 as amended herein above, the present invention is directed to a method and apparatus for instrumenting object oriented virtual-machine-executable software. As described and defined in

09/879,838

the present Application and as understood by those of ordinary skill in the arts, object oriented virtual-machine-executable software is comprised of class files containing bytecode wherein the bytecode in turn includes instructions defining the operation and functionality of the class files and wherein each class file is defined by a class and is associated with a class-hierarchy location and has a corresponding class name.

According to the present invention, the instrumentation processes create instrumented class files such that a virtual machine will process the instrumented class, thereby providing the desired instrumentation, but wherein the instrumentation is implemented such that the instructions of the original target class remain unmodified.

According to the present invention, a target class is instrumented by an instrumentation process that is selected from a set of three related instrumentation processes wherein the specific process that is to be used in a given instance is dependent upon the circumstances and the type of instrumentation to be provided.

In a presently preferred embodiment of the invention, the first instrumentation process is defined and implemented by the steps of (a) creating a new instrumented class separate from the target class, (b) adding instrumentation to the new instrumented class, and (c) assigning the new instrumented class to a class-hierarchy location adjacent to and above the first class-hierarchy location of the target class. The instrumentation is thereby comprised of additions to the bytecode of the newly created class rather than as modifications to the instructions of the original target class, so that the instructions of the target class remain in their original, unmodified form but are instrumented by means of the instrumentation added to the newly created class that is effectively located in the execution path(s) leading to the target class. It must also be noted that the new instrumented class is not a modified version of the original target class, but is an entirely separate class created independently and separately from the target class for the specific purpose of providing instrumentation for the target class.

The second instrumentation process is defined and implemented by the steps of (a) creating a new instrumented class separate from the target class, (b) adding instrumentation to the new instrumented class, (c) assigning the new instrumented class to a class-hierarchy location adjacent to and below the first class-hierarchy location of the target class, (d) assigning the first class name to the new instrumented class and (e) assigning a second class name to the target class. The instrumentation is again comprised of additions to the bytecode of the newly created

09/879,838

class rather than as modifications to the instructions of the original target class, so that the instructions of the target class again remain in their original, unmodified form but the target class is instrumented by means of the instrumentation added to the newly created class that is effectively located in the execution path(s) leading from the target class. It must again be noted that the new instrumented class is not the original target class or a modified version of the original target class, but is an entirely separate class created independently and separately from the target class for the specific purpose of providing instrumentation for the target class.

The third instrumentation process is defined and implemented by the steps of (a) adding instrumentation to the bytecode of the target class without modifying the instructions of the target class. The instrumentation is thereby implemented as modifications to the bytecode of the original target class, but the instructions of the target class again remain in their original, unmodified form.

The Applicant refers the Examiner to Figs. 9 through 12 and pages 32 through 44 of the Specification for detailed discussions of the above described instrumentation method of the present invention and the three instrumentation processes included in the instrumentation method, and for discussions of the reasons and results of the method of the present invention. It will be noted, moreover, that Figs. 1 through 8 and the preceding portions of the Specification describe in detail how and why the methods of the class files are invoked in object oriented virtual machine executable software, thereby providing the background for understanding of the present invention.

It will be noted that the Applicant has amended the claims herein above, and in particular claims 1 and 13, to clarify and more explicitly recite the above described invention.

Now considering the rejection of claims 1 and 8-13 are rejected under 35 U.S.C. § 103(a) over Angel et al. '558 in view of Berry et al. '237, the present invention as recited in independent claims 1 and 13 is fully and patentably distinguished over and from the teachings of both Angel et al. '558 and Berry et al. '237 for a number of fundamental reasons.

For example, and first considering Angel et al. '558, it has been described above and is recited in claims 1 and 13 that the method and apparatus of the present invention instruments a target class by (a) creating a new instrumented class that is separate and independent from the target class rather than by modifying the existing target class and (b) inserting the new instrumented class in the path into or out of the target class or in replacement for the target class while the target class remains unmodified or (c) by adding instrumentation to the target class without modifying the instructions of the target class.

09/879,838

In basic contrast from the present invention, Angel et al. '558 clearly teaches the instrumentation of a class by modifying the bytecode instructions of the class to be instrumented, so that Angel et al. '558 thereby (a) does not create a new instrumented class that is separate and independent from the class to be instrumented and (b) clearly teaches and requires the modification of the instructions of the class to be instrumented. These teachings of Angel et al. '558, and the resulting distinctions from the present invention, are clearly and completely described in Angel et al. '558 in, for example, the Abstract. and at column 3, lines 16 through 33, of the Specification. It must also be noted that these teachings of Angel et al. '558 are further fully supported and confirmed in the more detailed descriptions of Angel et al. '558 appearing for example in column 20, line 44, through column 32, line 50.

Further in this regard, it must be noted that for the above reasons the Applicant respectfully disagrees with certain of the Examiner's interpretations of the teachings of Angel et al. '558 as stated by the Examiner in the present Office Action. For example, and referring to item (iii) on page 4 of the Action, it is the Applicant's position that Angel et al. '558 does not teach or even suggest "adding instrumentation to the target class without modifying bytecode within the target class" (col. 3, lines 22-23). In contrast, Angel et al. '558 clearly describes that the instrumentation is accomplished by modifying the bytecode/instructions of the target class. Col. 3, lines 22-23 of the Angel et al. '558 specification merely refers to "[i]nstrumenting a portion of the byte code corresponding to a method call", which, placed in context, clearly states that the target class is instrumented by instrumenting a portion of the instructions of the target class, that is, by modifying the method call, and, just as clearly, does not state how that instrumenting is actually performed. As discussed, however, other portions of the Angel et al. '558 Specification clearly state that the instrumentation of a class is accomplished by modifying the bytecode of the class, that is, by modifying the instructions of the class.

It is also noted that the Examiner refers in section (c) on page 5 of the Action to "causing a virtual machine to process as the target class the class assigned the first class name" and refers to col. 20, lines 62-26, of the Angel et al. '558 Specification wherein it is stated that "Byte code may be instrumented by instrumenting each class as the class is loaded by the VM runtime system". The Applicant does not understand what point the Examiner is making in section (c) on page 5 of the Action as the two statements refer to entire different operations. The first refers to causing a virtual machine to process a class assigned the first class name as the target class, which is the

09/879,838

method used by the present invention to cause the virtual machine to process the instrumented class in place of the target class when the virtual machine is processing the software. The second statement merely states that the bytecode, that is, the instructions, of a class file may be instrumented when the class is loaded for execution, which is an entirely different operation from tricking the virtual machine to process a first class in place of a second class by giving the first class the name of the second class. Stated another way, the second statement is directed to when something is done, and not to how it is done.

As such, it is clear that the second statement, from col. 20, lines 62-26, of the Angel et al. '558 Specification that "Byte code may be instrumented by instrumenting each class as the class is loaded by the VM runtime system" does not relate at all to the method of the present invention of either instrumenting a new instrumented class that is separate and independent from the target class, which remains unmodified and in its original form, and tricking the virtual machine into invoking the new instrumented class rather than the unmodified target class, or of adding instrumentation to the target class without modifying the instructions of the target class.

It is therefore the belief and position of the Applicant that, for the reasons discussed above, Angel et al. '558 does not teach, suggest or hint at fundamental and essential aspects of the present invention that are recited in claims 1 and 13. Angel et al. '558, in fact, teaches directly away from the present invention by explicitly teaching the direct opposite of the present invention. That is, Angel et al. '558 teaches modifying the original bytecode of the target class, that is, modifying the instructions of the target class, in order to instrument the target class while the present invention is directed to a method that explicitly does not modify the instructions of the target class. That is, and in further detail, the process of the present invention instead either creates a new instrumented class that is separate and independent from the target class and inserts the new instrumented class into the software in such as way as to cause the virtual machine to invoke the new instrumented class rather than the original and unmodified target class, thereby leaving the instructions of the target class unmodified, or by inserting the instrumentation into the target class without modifying the instructions of the target class.

It is therefore the belief and position of the Applicant that independent claims 1 and 13 are fully and patentably distinguished over and from the teachings and suggestions of Angel et al. '558 under the requirements and provisions of 35 U.S.C. 103 and that Angel et al. '558 in fact teaches directly away from the present invention as recited in independent claims 1 and 13.

09/879,838

Next considering the teachings of Berry et al. '237, this reference is similar to Angel et al. '558, and in basic contrast from the present invention, in clearly teaching the instrumentation of a class by modifying the bytecode (instructions) of the class to be instrumented, so that Berry et al. '237 (a) does not create a new instrumented class that is separate and independent from the class to be instrumented and (b) clearly teaches and requires the modification of the bytecode, that is, the instructions, of the class to be modified. Such teaching of Berry et al. '237, and the resulting distinctions from the present invention, are clearly and completely described in Berry et al. '237 in, for example, the Abstract, in Fig. 2 and at column 3, line 57 through column 4, line 27 of the Specification.

It must also be noted that these teachings of Berry et al. '237 are further fully supported and confirmed in the more detailed descriptions of Berry et al. '237 appearing for example in columns 7 through 9. For example, it is noted that the Examiner cites col. 5, lines 20-24 of Berry et al. '237 in a subsequent rejection of claims 8, 9 and 10 and that col. 5, lines 20-24 of Berry et al. '237 state that "[c]ode may be added to, deleted from, or modified in the Java class file for many reasons, including instrumentation, benchmarking, performance tuning, modifying functionality, applying functional or performance patches". It is clear from this statement by Berry et al. '237 that Berry et al. '237 does in fact teach the modification of the instructions of an existing file that is to be instrumented and does not teach or even suggest or imply either the creation of a new file containing the instrumentation or of inserting the instrumentation without modifying the instructions of the target class to be instrumented.

It must also be noted with regard to the above that the Applicant respectfully disagrees with the Examiner's interpretation of the teachings of Berry et al. '237 as expressed on page 5 of the Action. For example, the Examiner cites Berry et al. '237 at col. 7, line 15, wherein Berry et al. '237 states "[t]he modified class file is then created (step 522)", and at col. 5, line 21, wherein Berry et al. '237 states "adding code for the purposes of instrumentation", and interprets these statements as saying that Berry et al. '237 creates a separate new class and adds instrumentation to the separate new class.

The Examiner therefore appears to have interpreted the modification of an existing class file as comprising the creation of a separate file. These statements by Berry et al. '237 are, in fact, more properly interpreted as stating that Berry et al. '237 modifies the class file to create a modified version of the class file, wherein the modification is the addition of the instrumentation by

09/879,838

the modification of the instructions of the class file, rather than as stating that Berry et al. '237 creates an entire new file separate and distinct from the original class file, as is done in the present invention. It must be further noted that this interpretation, that is, that Berry et al. '237 modifies an existing class file to create an instrumented class rather than creating a new instrumented class file that is separate and distinct from the original class file, is in full accordance with the remaining portions of the Berry et al. '237 Specification and drawings.

It is therefore the belief and position of the Applicant that, for the reasons discussed above, Berry et al. '237 does not teach, suggest or hint at fundamental and essential aspects of the present invention that are recited in claims 1 and 13. Berry et al. '237 in fact teaches directly away from the present invention by explicitly teaching the direct opposite of the present invention, that is, Berry et al. '237 teaches modifying the original byte code (instructions) of the target class in order to instrument the target class.

In fundamental contrast from Berry et al. '237, the present invention is directed to (a) a method that explicitly does not modify the instructions of the target class, that (b) in two of the three methods within the process creates a new instrumented class that is separate and independent from the target class and inserts the new instrumented class into the software in such a way as to cause the virtual machine to invoke the new instrumented class rather than the original and unmodified target class, and that (c) in the third method of the process adds the instrumentation to the target class without modifying the instructions of the target class.

It is therefore the belief and position of the Applicant that independent claims 1 and 13 are fully and patentably distinguished over and from the teachings and suggestions of Berry et al. '237 under the requirements and provisions of 35 U.S.C. 103 and that Berry et al. '237, in fact, teaches directly away from the present invention as recited in independent claims 1 and 13.

Therefore considering the teachings and suggestions of Angel et al. '558 in view of Berry et al. '237, it is clear that both Angel et al. '558 and Berry et al. '237 clearly teach the instrumentation of a class by modifying the bytecode (instructions) of the class to be instrumented, so that neither Angel et al. '558 nor Berry et al. '237 teaches or suggest (a) the creation of a new instrumented class that is separate and independent from the class to be instrumented and that both Angel et al. '558 and Berry et al. '237 clearly teach and require (b) the modification of the bytecode (instructions) of the class to be instrumented.



09/879,838

For these reasons, therefore, it is the belief and position of the Applicant that, for the reasons discussed above, neither Angel et al. '558 nor Berry et al. '237 teaches, suggests or hints at fundamental and essential aspects of the present invention that are recited in claims 1 and 13. Angel et al. '558 and Berry et al. '237 both, in fact, teach directly away from the present invention by explicitly teaching the direct opposite of the present invention. That is, Angel et al. '558 and Berry et al. '237 teach modifying the original byte code (instructions) of the target class in order to instrument the target class while the present invention is directed to a method that explicitly does not modify the instructions of the target class and that, in two of the process methods, further creates a new instrumented class that is separate and independent from the target class and inserts the new instrumented class into the software in such as way as to cause the virtual machine to invoke the new instrumented class rather than the original and unmodified target class.

It is therefore the belief and position of the Applicant that independent claims 1 and 13 are fully and patentably distinguished over and from the teachings and suggestions of Angel et al. '558 and Berry et al. '237 and Angel et al. '558 in view of Berry et al. '237 under the requirements and provisions of 35 U.S.C. 103 and that Angel et al. '558 and Berry et al. '237 and Angel et al. '558 in view of Berry et al. '237, in fact, teach directly away from the present invention as recited in independent claims 1 and 13. The Applicant therefore respectfully requests that the Examiner reconsider and withdraw all rejections of independent claims 1 and 13 as amended herein above under 35 U.S.C. 103 over Angel et al. '558 and Berry et al. '237 and Angel et al. '558 in view of Berry et al. '237, and the allowance of claims 1 and 13 as amended herein above.

Next considering the rejection of claims 8, 9 and 10 under 35 U.S.C. § 103(a) over Angel et al. '558 in view of Berry et al. '237, the Examiner states that Angel et al. '558 does not teach a virtual machine that is a Java virtual machine and the addition of instrumentation to or modification of the target class by modification of one of the class components recited in claim 8 or by the modification of memory locations as recited in claims 9 and 10. The Examiner does state, however, that Berry et al. '237 describes a Java virtual machine and such modifications to a class file.

In response, it must be noted that the present invention is not directed to these methods for modifying class files per se, but to a method and apparatus that employs such methods to instrument a target class file by the addition of instrumentation without modification of the class file instructions and by creating a new instrumentation file that is separate and independent from the

09/879,838

target class and placing the instrumentation in the new instrumentation file while leaving the original target class unmodified. The methods for class file modification recited in claims 8, 9 and 10 are used to effectively redirect the virtual machine away from the unmodified target class and to the new created instrumented class. That is, these methods are employed, according to the present invention, to cause the newly created instrumented class to appear to the virtual machine as the original target class while causing the virtual machine to not recognize the original target class as the original target class.

In contrast from the methods and apparatus of the present invention appearing in claims 8, 9 and 10, the methods taught by Berry et al. '237 are employed by Berry et al. '237 to modify the "target" class by modification of the instructions of the target class to thereby directly instrument the target class files, rather than, as in the present invention, to instrument the target class without modification of the instructions of the target class and to create a new instrumented class and to cause the newly created instrumented class to appear to the virtual machine as the original target class while causing the virtual machine to not recognize the original target class as the original target class. Therefore, while the class file modification methods taught by Berry et al. '237 are somewhat analogous to those used in the present invention as recited in claims 8, 9 and 10 to redirect the virtual machine, the use and purpose of the methods of the present invention are fundamentally distinguished from the uses and purposes of those methods by Berry et al. '237.

As such, it is the belief and position of the Applicant that although the specific steps within the methods of the present invention as recited in claims 8, 9 and 10 are similar to those employed in the class file modification methods used by Berry et al. '237, the methods of the present invention are fully distinguished from the class file modification methods for Java virtual machines as taught by Berry et al. '237 by being directed to fundamentally different purposes and results.

In addition, and further fundamental distinction between the present invention and the teachings of Berry et al. '237, it must be noted that claims 8, 9 and 10 are dependent from claim 1 and thereby incorporate by dependence all recitations and limitations of claim 1. For the reasons discussed herein above, claim 1 is fully and patentably distinguished over and from Angel et al. '558 and Berry et al. '237 and the combination of Angel et al. '558 in view of Berry et al. '237 under the requirements and provisions of 35 U.S.C. 103. For the same reasons, therefore, and because the recitations and limitations of claim 1 are incorporated into claims 8, 9 and 10 by dependence therefrom, claims 8, 9 and 10 are thereby also fully and patentably distinguished over

09/879,838

and from Angel et al. '558 and Berry et al. '237 and the combination of Angel et al. '558 in view of Berry et al. '237 under the requirements and provisions of 35 U.S.C. 103.

The Applicant therefore respectfully requests that the Examiner reconsider and withdraw the rejections of claim 8, 9 and 10 under 35 U.S.C. 103 over Angel et al. '558 and Berry et al. '237 and the combination of Angel et al. '558 in view of Berry et al. '237, and the allowance of claims 8, 9 and 10.

Next considering the rejection of claims 11 and 12 under 35 U.S.C. § 103(a) over Angel et al. '558 in view of Berry et al. '237, the Examiner states that Angel et al. '558 has no teachings relevant to the recitations of claims 11 and 12 but cites Berry et al. '237 at col. 1, lines 60-62 as teaching superclasses as recited in claims 11 and 12.

The Applicant respectfully disagrees with the Examiner's interpretation of the relationship of the teachings of Berry et al. '237 regarding superclasses and the recitations of claims 11 and 12. The cited statement in Berry et al. '237 is in the Background of the Invention and is part of a generalized description of the well known and commonly used object oriented Java class files and their hierarchical relationship to one another, such as the relationship of a class to a "superclass" of which it is a member. As such, the description of Java class files, classes and superclasses is no more than background information and has little relevance to various actual uses of such class/superclass relationships in the present invention.

For example, the descriptions of Java file classes and superclasses in Berry et al. '237 does not describe or even suggest the use, as in the present invention as recited in claims 11 and 12, of class and superclass relationships to insert a new instrumented class in the execution/invocation path to or from a target class to thereby instrument the unmodified target class by means of the new instrumented class. Berry et al. '237 instead describes only the well known and long established organizational and structural arrangements for object oriented code, rather than describing the use of such organizational and structural restrictions for new purposes not set forth in the original, basic structural restrictions imposed in object oriented programming for other reasons.

As such, it is the belief and position of the Applicant that the cited teachings of Berry et al. '237, while pertinent to Java class files, classes and superclasses, is not relevant to the specific aspects of the present invention that are recited in claims 11 and 12.

-18-

09/879,838

In addition, and in fundamental distinction between the present invention and the teachings of Berry et al. '237 and of Angel et al. '558 in view of Berry et al. '237, it must be noted that claims 11 and 12 are dependent from claim 1 and thereby incorporate by dependence all recitations and limitations of claim 1.

For the reasons discussed herein above, claim 1 is fully and patentably distinguished over and from Angel et al. '558 and Berry et al. '237 and the combination of Angel et al. '558 in view of Berry et al. '237 under the requirements and provisions of 35 U.S.C. 103. For the same reasons, therefore, and because the recitations and limitations of claim 1 are incorporated into claims 11 and 12 by dependence therefrom, claims 11 and 12 are thereby also fully and patentably distinguished over and from Angel et al. '558 and Berry et al. '237 and the combination of Angel et al. '558 in view of Berry et al. '237 under the requirements and provisions of 35 U.S.C. 103.

The Applicant therefore respectfully requests that the Examiner reconsider and withdraw the rejections of claim 11 and 12 under 35 U.S.C. 103 over Angel et al. '558 and Berry et al. '237 and the combination of Angel et al. '558 in view of Berry et al. '237, and the allowance of claims 11 and 12.

Next considering the Examiner's rejections of claims 2-7 and 14 under 35 U.S.C. § 103(a) over Angel et al. '558 in view of Bryant et al. '367, and in particular the rejection of claims 2, 3, 5 and 6, the Examiner again cites the teachings of Angel et al. '558 but also states with regard to claims 2, 3, 5 and 6 that Angel et al. '558 does not teach the operations of suspending the operation of a virtual machine between the loading and linking of a target class.

The Examiner further states, however, that Bryant et al. '367, in brief, does contain such teachings and cites Bryant et al. '367 at col. 6, lines 56-57 and 57-60.

As discussed above with regard to the Examiner's citing of Angel et al. '558 with regard to the alleged teaching of the non-modification of the target class by Angel et al. '558, the citing of a few lines from a specification and out of context can result in a misinterpretation of the actual teachings of the cited reference. The actual meaning of a prior art reference can be fully understood only when the entire reference and all parts thereof are taken in context and, in this instance, the Applicant again respectfully disagrees with the Examiner's interpretation of the cited reference.

In particular, Bryant et al. '367 teaches a method for increasing the performance of the execution of a Java application on a Java server by using Java application code and functions of

09/879,838

the application on the server as a library of classes and methods. When the application is to be executed, an object file calls the appropriate Java server process of the application on the Java server and the Java server process forks itself as a child server from the already existing parent server in the conventional manner and causes the child server to run the already loaded classes and methods of the application code. That is, the object calling the parent Java server communicates with the parent Java server while, in actuality the parent Java server has forked the child Java server, which is executing the methods and classes of the parent Java server and exchanging calls and returns, including data, between the child and parent Java servers. It therefore appears to the calling object that the parent Java server is executing the methods and classes of the called process while, in actuality, it is the child server that is executing the methods and classes.

As a consequence, and as specifically taught by Bryant et al. '367, it is only necessary to load the classes and methods once, that is, as the original parent server when the Java virtual machine is started. According to Bryant et al. '367, this method enhances the speed of execution of a Java process because it is faster to fork an already running Java server to a child server in order to execute the necessary classes and methods than it is to load the Java virtual machine, that is, the parent Java server, at each execution of the Java process.

A complete reading of Bryant et al. '367 therefore shows that the cited portions of the Bryant et al. '367 Specification in fact describe the interaction between the parent Java server and the child Java server and does not relate or pertain to the structural or operational relationships between class files within a Java process executing on a single virtual machine, or server. As such, it is apparent that the teachings of Bryant et al. '367 are in fact not material or relevant to the present invention as recited in claim 1 or in claims 2, 3, 5 and 6.

In addition, and in fundamental distinction between the present invention and the teachings of Bryant et al. '367, Angel et al. '558, Berry et al. '237 and Bryant et al. '367 and of Angel et al. '558 in view of Berry et al. '237 in view of Bryant et al. '367, it must be noted that claims 2, 3, 5 and 6 are dependent from claim 1 and thereby incorporate by dependence all recitations and limitations of claim 1. In particular, none of Bryant et al. '367, Angel et al. '558 or Berry et al. '237 teach, suggest or hint at fundamental and essential aspects of the present invention that are recited in claim 1. Angel et al. '558 and Berry et al. '237, in fact, teach directly away from the present invention by explicitly teaching the direct opposite of the present invention, that is, by teaching

09/879,838

modification of the original byte code (instructions) of the target class in order to instrument the target class, while Bryant et al. '367 is not material or pertinent to the present invention as recited in claim 1. As discussed, the present invention as recited in claim 1 is directed to a method that explicitly does not modify the instructions of the target class and that for certain instances creates a new instrumented class that is separate and independent from the target class and inserts the new instrumented class into the software in such a way as to cause the virtual machine to invoke the new instrumented class rather than the original and unmodified target class.

For the reasons discussed herein above, therefore, claim 1 is fully and patentably distinguished over and from Angel et al. '558 and Berry et al. '237 and Bryant et al. '367 and over and from all combinations of Angel et al. '558 in view of Berry et al. '237 in view of Bryant et al. '367 under the requirements and provisions of 35 U.S.C. 103. For the same reasons, therefore, and because the recitations and limitations of claim 1 are incorporated into claims 2, 3, 5 and 6 by dependence therefrom, claims 2, 3, 5 and 6 are thereby also fully and patentably distinguished over and from Angel et al. '558 and Berry et al. '237 and Bryant et al. '367 and all combinations of Angel et al. '558 in view of Berry et al. '237 in view of and Bryant et al. '367 under the requirements and provisions of 35 U.S.C. 103.

The Applicant therefore respectfully requests that the Examiner reconsider and withdraw the rejections of claim 2, 3, 5 and 6 under 35 U.S.C. 103 over Angel et al. '558 and Berry et al. '237 and Bryant et al. '367 and all combinations of Angel et al. '558 in view of Berry et al. '237 in view of Bryant et al. '367, and the allowance of claims 2, 3, 5, and 6.

Next considering the Examiner's rejections of claims 4 and 7 under 35 U.S.C. § 103(a) over Angel et al. '558 in view of Bryant et al. '367, as has been discussed herein above with regard to the rejection of claim 1, none of Bryant et al. '367, Angel et al. '558 or Berry et al. '237 teach, suggest or hint at fundamental and essential aspects of the present invention that are recited in claim 1. Angel et al. '558 and Berry et al. '237 both, in fact, teach directly away from the present invention by explicitly teaching the direct opposite of the present invention, that is, by teaching modification of the original byte code (instructions) of the target class in order to instrument the target class, while Bryant et al. '367 is not material or pertinent to the present invention as recited in claim 1. As discussed, the present invention as recited in claim 1 is directed to a method that explicitly does not modify the instructions of the target class and that in certain instances creates a new instrumented class that is separate and independent from the target class and inserts the

09/879,838

new instrumented class into the software in such as way as to cause the virtual machine to invoke the new instrumented class rather than the original and unmodified target class.

For the reasons discussed herein above, therefore, claim 1 is fully and patentably distinguished over and from Angel et al. '558 and Berry et al. '237 and Bryant et al. '367 and over and from all combinations of Angel et al. '558 in view of Berry et al. '237 in view of Bryant et al. '367 under the requirements and provisions of 35 U.S.C. 103. For the same reasons, therefore, and because the recitations and limitations of claim 1 are incorporated into claims 4 and 7 by dependence therefrom, claims 4 and 7 are thereby also fully and patentably distinguished over and from Angel et al. '558 and Berry et al. '237 and Bryant et al. '367 and all combinations of Angel et al. '558 in view of Berry et al. '237 in view of and Bryant et al. '367 under the requirements and provisions of 35 U.S.C. 103.

Next considering the teachings of Bryant et al. '367 with respect to claims 4 and 7, the Examiner cites Bryant et al. '367, and in particular at col. 26, lines 21-16, at the remainders of cols. 26 and 27, and at col. 7, lines 45-53, at teaching the elements of claims 4 and 7 that are not taught by Angel et al. '558.

The Applicant again respectfully disagrees with the Examiner's interpretation of the teachings of Bryant et al. '367 on the grounds that the cited teachings are extracted from the context of the Bryant et al. '367 Specification and do not consider the entirety of the Bryant et al. '367 teachings.

In particular, and as discussed above, Bryant et al. '367 teaches a method for increasing the performance of the execution of a Java application on a Java server by using Java application code and functions of the application on the server as a library of classes and methods. When the application is to be executed, an object file calls the appropriate Java server process of the application on the Java server and the Java server process forks itself as a child server from the already existing parent server in the conventional manner and causes the child server to run the already loaded classes and methods of the application code. That is, the object calling the parent Java server communicates with the parent Java server while, in actuality the parent Java server has forked the child Java server, which is executing the methods and classes of the parent Java server and exchanging calls and returns, including data, between the child and parent Java servers. It therefore appears to the calling object that the parent Java server is executing the

09/879,838

methods and classes of the called process while, in actuality, it is the child server that is executing the methods and classes.

A complete reading of Bryant et al. '367 shows that the cited portions of the Bryant et al. '367 Specification refer to the method by which the child server identifies and locates the required classes and methods in the parent server, which as described is employed as a library of classes and methods used in the child server process, which as described is executed in place of the server process residing in the parent server. Stated another way, Bryant et al. '367 describes a method by which one server, that is, the child server, can identify and locate a plurality of classes and methods residing in a second server, that is, the parent server.

The operation described by Bryant et al. '36 is therefore fundamentally and significantly distinguished from the operations of the present invention as recited in claims 4 and 7, which pertains to a method for identifying a specific single target class in a Java virtual machine by searching for and matching a specified set of attribute names and associated values within a single Java virtual machine and a single process. In fundamental contrast from the present invention as recited in claims 4 and 7, the method taught by Bryant et al. '367 pertains to the identification and location by one process running on one virtual machine of methods and classes of a library of methods and classes residing in another process in another virtual machine.

As such, it is apparent that the teachings of Bryant et al. '367 are in fact not material or relevant to the present invention as recited in claim 1 or in claims 4 and 7. The Applicant therefore respectfully requests that the Examiner reconsider and withdraw the rejections of claims 4 and 7 under 35 U.S.C. 103 over Angel et al. '558 and Bryant et al. '367 and Berry et al. '237 and all combinations of and all combinations of Angel et al. '558 in view of Bryant et al. '367 in view of Berry et al. '237, and the allowance of claims 4 and 7.

Referring finally to the rejection of claim 14 under 35 U.S.C. § 103(a) over Angel et al. '558 in view of Bryant et al. '367, it will first be noted that the above submitted amendments to the claims have made apparatus claim 14 dependent from apparatus claim 13, which is essentially directed to the same invention and contains the same recitations and limitations as method claim 1. As discussed herein above, claim 13 is fully and patentably distinguished over and from the teachings and suggestions of Angel et al. '558, Berry et al. '237 and Bryant et al. '367 and all combinations of Angel et al. '558, Berry et al. '237 and Bryant et al. '367 under the requirements and provisions of 35 U.S.C. 103.

-23-



09/879,838

Claim 14, like claim 4, is directed to an apparatus for identifying a target class included within the virtual machine executable software according to the purposes, functions and recitations of claim 13 and, being dependent from claim 13, incorporates all recitations and limitations of claim 13 by that dependence. Claim 14 is thereby fully and patentably distinguished over and from the teachings and suggestions of Angel et al. '558, Berry et al. '237 and Bryant et al. '367 and all combinations of Angel et al. '558, Berry et al. '237 and Bryant et al. '367 under the requirements and provisions of 35 U.S.C. 103 for the same reasons that claim 13 is fully and patentably distinguished over and from the teachings and suggestions of Angel et al. '558, Berry et al. '237 and Bryant et al. '367 and all combinations of Angel et al. '558, Berry et al. '237 and Bryant et al. '367 under the requirements and provisions of 35 U.S.C. 103.

The Applicant therefore respectfully requests that the Examiner reconsider and withdraw the rejection of claim 14 under 35 U.S.C. 103 over Angel et al. '558 and Bryant et al. '367 and Berry et al. '237 and all combinations of and all combinations of Angel et al. '558 in view of Bryant et al. '367 in view of Berry et al. '237, and the allowance of claim 14 as amended herein above.

If any further amendment to this application is believed necessary to advance prosecution and place this case in allowable form, the Examiner is courteously solicited to contact the undersigned representative of the Applicant to discuss the same.

In view of the above amendments and remarks, it is respectfully submitted that all of the raised rejection(s) should be withdrawn at this time. If the Examiner disagrees with the Applicant's view concerning the withdrawal of the outstanding rejection(s) or applicability of the Angel et al. '558, Berry et al. '237 and/or Bryant et al. '367 references, the Applicant respectfully requests the Examiner to indicate the specific passage or passages, or the drawing or drawings, which contain the necessary teaching, suggestion and/or disclosure required by case law. As such teaching, suggestion and/or disclosure is not present in the applied references, the raised rejection should be withdrawn at this time. Alternatively, if the Examiner is relying on his/her expertise in this field, the Applicant respectfully requests the Examiner to enter an affidavit substantiating the Examiner's position so that suitable contradictory evidence can be entered in this case by the Applicant.

In view of the foregoing, it is respectfully submitted that the raised rejection(s) should be withdrawn and this application is now placed in a condition for allowance. Action to that end, in the form of an early Notice of Allowance, is courteously solicited by the Applicant at this time.

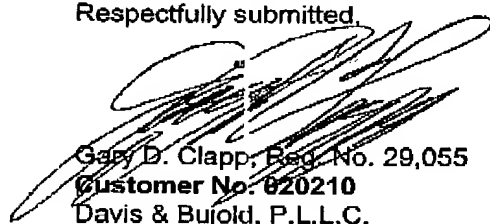
-24-

09/879,838

The Applicant respectfully requests that any outstanding objection(s) or requirement(s), as to the form of this application, be held in abeyance until allowable subject matter is indicated for this case.

In the event that there are any fee deficiencies or additional fees are payable, please charge the same or credit any overpayment to our Deposit Account (Account No. 04-0213).

Respectfully submitted,



Gary D. Clapp, Reg. No. 29,055  
Customer No. 020210  
Davis & Bujold, P.L.L.C.  
Fourth Floor  
500 North Commercial Street  
Manchester NH 03101-1151  
Telephone 603-624-9220  
Facsimile 603-624-9229  
E-mail: patent@davisandbujold.com

-25-